REMARKS

In view of the above amendments and the following remarks, reconsideration and further examination are requested.

Minor editorial amendments have been made to the specification and abstract. No new matter has been added. A substitute abstract along with a marked-up copy of the substitute abstract are filed herewith.

Claims 1, 3, 5-11, and 19 were rejected under 35 USC 102(e) as being anticipated by Farmer. This rejection is traversed and is also inapplicable to new claims 20-27 for the following reasons.

Independent claims 1 and 5 have been canceled in favor of new claims 20 and 23, respectively.

Each of independent claims 20, 23, and 19, includes a recitation directed to a second-endian processor (i.e., a big-endian or little-endian processor) logically connected to a memory in a first-endian byte order. Claims 20 and 23 also include a recitation of an address conversion unit, (and method claim 19 includes a recitation of performing an address conversion), wherein an address conversion is performed on at least one lower bit of an address so as to indicate a reversed position of data in the data bus when the processor performs a <u>memory access for data having a smaller width</u> than the width of the data bus, and wherein the address conversion is not performed when the processor performs a <u>memory access for data having the width of the data bus</u>.

Thus, the apparatuses and methods recited in claims 20, 23, and 19 enable a big-endian processor and a little-endian processor to be logically connected to the memory with the same byte order (i.e., a big-endian byte order or a little-endian byte order). When memory access for data having the width of the data bus is performed, the address conversion is not needed and is therefore not performed. In such a case, both processors perform the memory access via the same byte order, e.g., both a big-endian and a small-endian processor perform memory access in a big-endian byte order. However, when the memory access is performed for data having a width smaller than the data bus, the address conversion is performed on the lower bit or bits of the address so that a reversed position of data is indicated.

For example, assume a data bus having a width of 32 bits, and a big-endian processor and little-endian processor logically connected to the memory in a little-endian byte order. In such an example, if memory access is performed for 32 bit data, then no address conversion is performed and the big-endian processor performs the memory access in the little-endian byte order with no problem, even though such a byte order is opposite of the processor's normal connection. If memory access is performed for data having a width smaller than 32 bits, one or more lower bits of the address is converted depending on the size of the data. For example, for 16 bit data one bit of the address is converted, whereas for 8 bit data two bits are converted.

The Farmer reference does not disclose or suggest a second-endian processor logically connected to a memory in a first-endian byte order, performing address conversion of at least one lower address bit when memory access is performed for data having a smaller width than the data bus, and not performing the address conversion when memory access is performed for data having the width of the data bus.

The system of Farmer includes a big-endian processor 210, a little-endian processor 220, an interface 230, and a memory 240 in the big-endian format (see, e.g., Fig. 2 of Farmer). The processors are connected to the memory via the interface 230. The system of Farmer uses a signal, for example, an Endian Alias Bit (EAB) which indicates the endianness of the processor (see column 2, lines 60-65, and column 3, lines 34-37). The interface 230 (using its multiplexers 320) selectively reconfigures the data depending on whether the EAB is set (see column 3, lines 19-21).

Farmer does not disclose or in any way suggest not performing the reconfiguration when memory access is performed for data having the width of the data bus. The interface 230 of Farmer always reconfigures the data for the little-endian processor because the memory 240 is in the big-endian format. Thus, for the little-endian processor of Farmer, the EAB indicates that reconfiguration by the interface 230 is necessary irrespective of the access size of the data. On the contrary, in the present invention as recited in claims 20, 23, and 19, when the data is the width of the data bus, conversion is not needed and is not performed because the second-endian processor (e.g., little-endian) is logically connected to the memory in the first-endian byte order (e.g., big-endian byte order).

Because of the distinctions discussed above, independent claims 20, 23, and 19, as well as the remaining claims 2-4, 6-18, 21, 22, and 24-27 are not anticipated by Farmer.

Claims 2, 4, and 12-18 were rejected under 35 USC 103(a) as being unpatentable over Farmer in view of Heeb. This rejection is traversed, and is also inapplicable to the claims as amended.

Heeb does not provide teaching of the features discussed above, and therefore, no combination of Heeb and Farmer would result in or otherwise render obvious the apparatuses and methods recited in claims 2-4, and 6-27 of the present application.

The Examiner relies on Heeb as showing a cache memory connected to the data bus in a byte order of the processor. However, as discussed in detail above, claims 20, 23, and 19 recite that the second-endian processor is logically connected to the memory in the first-endian byte order, whereas claims 2, 4, and 12 additionally recite a cache memory logically connected to the memory in the second-endian byte order. Heeb discloses that the processor 10 is entirely little endian, the bus 14 is little endian, and the data cache 16 is little endian. Heeb discloses that when the processor stores to a big endian region of the external memory 22, the byte converter 24 converts data (see column 4, lines 26-39). Thus, neither Heeb nor Farmer teaches logically connecting a little-endian (or big-endian) processor to a bus or memory in the opposite big-endian (or little-endian) byte order. Therefore, while Heeb discloses connecting a cache 16 to the processor in the byte order (little-endian) of the processor, it would not have been obvious to a person having ordinary skill in the art to connect a cache to a processor in the byte order of the processor and to connect the processor to a memory via a bus in the opposite-endian byte order.

In view of the above, it is submitted that claims 2-4, and 6-27 are allowable over the Farmer, Heeb, and any combination thereof.

Accordingly, it submitted that the present application is in condition for allowance. The Examiner is invited to contact the undersigned by telephone to resolve any remaining issues.

Respectfully submitted,

Kazutoshi FUNAHASHI et al.

By: ______________________
Jeffrey R. Filipek
Registration No. 41,471
Attorney for Applicants

JRF/fs
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
December 18, 2006

PATENT & TRADEMARK OFFICE OIPE IAP97 DEC 18 2006

Version with Markings to Show Changes Made

ABSTRACT OF THE DISCLOSURE

The data sharing apparatus in the present invention includes a first processor ~~10~~ and a second processor ~~20~~, each of a different endianness, that are both connected to the memory via the data bus, in a byte order based on the endianness of the first processor ~~10~~. It also includes an address conversion unit ~~21~~ which converts at least one lower bit of an address to indicate a reversed position of data in the data bus, and outputs the converted address to the memory, in the case where the second processor ~~20~~ performs a memory access on the shared memory for data with a smaller width than the data bus.